

## Список использованных источников

1. Методические указания «Промышленные датчики расхода». Челябинск, 2012.
2. Техническое описание «Промышленные датчики расхода». Челябинск, 2012.
3. Руководство по программированию «Программируемые логические контроллеры DVP-SS / SA / SX / ES / EX / EH», ред. от 15.08.2006.
4. Руководство по эксплуатации «DVP-SX», НПО "СТОИК ЛТД".

УДК 669-7

**Э. И. Гольфельд**

ФГАОУ ВПО «Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина», г. Екатеринбург, Россия

## БИБЛИОТЕКА SIGNALR 2.0 ДЛЯ ОПОВЕЩЕНИЯ КЛИЕНТОВ WEB-ПРИЛОЖЕНИЙ О СОБЫТИЯХ НА ПРОИЗВОДСТВЕ

### Аннотация

*Данная статья является обзором популярной библиотеки с открытым исходным кодом от компании Microsoft – SignalR 2.0, предназначенной для оповещений клиентов web-приложений о различных событиях, например, таких как обновление материалов сайта. Затрагиваются проблемы организации такого вида взаимодействия без применения SignalR. Сравнение различных способов приведено в таблице, что позволяет наглядно продемонстрировать недостатки классических подходов и показать все преимущества от использования библиотеки. В статье рассказывается как именно работает библиотека, а также особое внимание в статье уделено внутренней архитектуре библиотеки, поскольку без понимания устройства SignalR использование библиотеки может оказаться крайне неэффективным. Приведены базовые приемы работы с библиотекой. Таким образом, данную статью следует рассматривать как фундамент для дальнейшего изучения библиотеки.*

*Ключевые слова:* библиотека, оповещение, архитектура, взаимодействие, события.

### Abstract

*This article is a review of popular open source library from Microsoft – SignalR 2.0, which developed to notice clients of web applications for various events, for example, such as updating the site content. Addresses the organization problems of this type of interaction without using SignalR. Comparing the different methods are summarized in the table that allows demonstrating the shortcomings of classical approaches and showing the benefits of using the library. The article describes how the library works, and special attention is devoted to the internal architecture of the library, because without understanding the structure SignalR, use the library can be extremely inefficient. Article have basics examples of using the library. Thus, this article should be seen as a foundation for further study of the library.*

*Keywords:* signalR, library, Microsoft, architecture, events, web.

Развитие вычислительных средств служит толчком для повышения запросов пользователей, что приводит к постоянному появлению новых технологий. За период существования web-технологии прошли путь от статических текстовых страничек до сложных динамических web-приложений, работающих на сотнях серверов. В наши дни web-приложения нашли применение во многих областях: в индустрии развлечений, в науке, промышленности и образовании.

Особое внимание разработчики уделяли проблеме уведомления пользователей о каких-либо событиях. Раньше приходилось для получения и отображения обновленных данных на странице вручную производить перезагрузку всей страницы. Это создавало дискомфорт для пользователя и увеличивало нагрузку на сервер. В наши дни существует целый букет средств для реализации динамического оповещения пользователей о событиях, которые произошли на стороне сервера.

Во многих современных web-сервисах пользователь может наблюдать изменения на сайте в режиме реального времени и ему не приходится перезагружать страницу. В качестве примеров таких сервисов можно назвать социальные сети VK и Facebook. Сервисы, работающие по подобному сценарию, часто называют web-приложениями реального времени. Особенность таких приложений в том, что они преобразуют модель работы — запрос-ответ. В промышленности такой подход, например, позволяет автоматизировать сбор информации с датчиков. Кроме того, с применением таких технологий появляется возможность в режиме реального времени предоставлять информацию персоналу о ходе выполнения определенного технологического процесса. Область применения этих приложений бесконечна: в науке, образовании, экономике и т. д.

Достоинства и недостатки техник построения web-приложений реального времени представлены в таблице.

Как видно, на данный момент есть достаточно неплохой выбор технологий, но все они имеют какие-либо недостатки. Самыми тяжелыми, пожалуй, являются недостатки, связанные с совместимостью с тем или иным браузером. Это заставляет разработчиков использовать подходы, которые создают большую нагрузку на сервер.

Напрашивается самое очевидно решение – проверка браузеров, используемых пользователем, и в соответствии с этим возможность выбора той или иной оптимальной техники. Именно такой подход был реализован разработчиками библиотеки SignalR.

ASP.NET SignalR это библиотека для разработчиков ASP.NET, которая упрощает добавление функциональности, связанной с обновлением страниц в реальном времени.

SignalR предоставляет очень простой API для взаимодействия сервера с клиентами web-приложений и наоборот. Весь исходный код открыт и доступен на GitHub [<https://github.com/SignalR/>]. Кроме того, в разделе Wiki проекта есть много информации о SignalR и примеров. SignalR был придуман и реализован разработчиками Microsoft.

Как уже было сказано выше, SignalR самостоятельно определяет подходящую технику. Когда пользователь открывает страницу, работающую с SignalR, библиотека анализирует какие техники поддерживаются. Приоритет протоколов SignalR выглядит следующим образом (рисунок).



Рис. Приоритет протоколов SignalR

Архитектура SignalR очень проста, предоставляется два вида API: низкоуровневый и высокоуровневый (Hub API). Высокоуровневый API опирается на низкоуровневый. В большинстве случаев будет достаточно возможностей Hub API.

Сравнение техник уведомления клиентов

Техника	Описание	Преимущества	Недостатки
1	2	3	4
Polling	Постоянный опрос сервера Ajax-запросами	+Простота реализации +Поддержка во всех современных браузерах	- Задержка в результатах - При уменьшении задержки существенно увеличивается нагрузка на сервер
Long Polling	Ajax-запросы, идущие один за другим, но каждый запрос держится открытым в течение нескольких минут	+ Сниженная нагрузка на сервер по сравнению с обычным Polling + Уменьшенный трафик + Поддержка во всех современных браузерах	- Больше одновременно открытых соединений, т. к. каждый запрос живет дольше
Server-Sent Events	Новый стандарт HTML5, работающий поверх HTTP. Позволяет создавать долгоживущее соединение с сервером	+ Нет необходимости постоянно пересоединяться с сервером + Нет изменений на стороне сервера, поэтому работает на всех современных веб-серверах	- Не поддерживается в IE (даже в IE10) - Работает только в направлении сервер → клиент (на сервер можно отправлять обычные Ajax запросы)
WebSockets	Новый протокол, работающий поверх TCP на одном уровне с HTTP. Позволяет создавать двустороннее долгоживущее соединение с клиентом	+ Нет необходимости постоянно пересоединяться с сервером + Работает в двустороннем режиме	- Поддерживается не во всех веб-серверах (IIS8) - Поддерживается не во всех браузерах (в IE 7-9, Android)

Hub API предполагает, что в центре клиент-серверного взаимодействия находится созданный разработчиком хаб – специальный класс. Этот класс содержит в себе методы взаимодействия клиента и сервера.

Класс хаба может выглядеть следующим образом:

```

public class MainHub : Hub
{
    /// Рассылка сообщений всем
    public void SendAll(string message)
    {
        Clients.All.addNewMsg(message);
    }
    /// Отправить группе
    public void SendGroup(string groupName, string message)
    {
        Clients.Group(groupName).addNewMsg(message);
    }
    /// Присоединиться к группе
    public Task JoinGroup(string groupName)
    {
        return Groups.Add(Context.ConnectionId, groupName);
    }
}

```

Хабовое взаимодействие осуществляется следующим образом:

1. Клиент устанавливает соединение с хабом сервера.
2. Клиент может войти в определенную группу.
3. Клиент и сервер могут вызывать методы хаба.
4. В зависимости от реализации клиент или сервер выполняет вызванный метод.

Стоит отметить, что на сервере может быть много хабов.

Есть несколько способов вызвать метод на клиенте: по его имени, по его группе и на всех клиентах сразу.

При помощи Hub API возможна реализация следующих возможностей: клиент вызывает метод на сервере, сервер вызывает метод на клиенте/клиентах, передача состояния с клиента на сервер и обратно, определение соединения, отсоединения и присоединения клиентов, передача объектов, подписка на события, отписка от событий, вход и выход из групп.

Например, для того чтобы разослать всем клиентам сообщение о повышении температуры в котле, необходимо выполнить следующий код:

```

///Создаем подключение
HubConnection hubConnection = new HubConnection("kotell.corpanyname.corp");
///Создаем прокси для хаба с именем "MainHub"
IHubProxy hubProxy = hubConnection.CreateHubProxy("MainHub");
///Запускаем подключение
hubConnection.Start().Wait();
///Вызываем хабовый метод SendAll и передаем сообщение об изменении температуры
hubProxy.Invoke("SendAll", newTemperatureMsg).Wait();
///Закрываем подключение
hubConnection.Stop();

```

Далее, на стороне клиента новая информация каким-либо образом обрабатывается, например, просто выводится на экран:

```
<!--Подключение SignalR и автоматически сгенерированного хаба SignalR -->
<script src="~/Scripts/jquery.signalR-2.0.0.min.js"></script>
<script src="~/signalr/hubs"></script>
<!--Скрипт для обработки событий-->
<script>
    $(function ()
    {
        // Создаем прокси для хаба "MainHub"
        var proxy = $.connection.MainHub;
        // Создаем функцию для обработки события "addNewMsg"
        proxy.client.addNewMsg = function (message)
        {
            // Добавляем сообщение на страницу
            $('#temperatureHistory').append('<li>' + message + '</li>');
        };
        // Запускаем соединение
        $.connection.hub.start().done()
    });
</script>
```

Таким образом, мы рассмотрели назначение, устройство и возможности библиотеки SignalR, рассмотрели способы построения приложений реального времени для web без применения библиотеки и выяснили, почему эти подходы плохи. Мы выяснили, как именно библиотека может использоваться в промышленности. Были продемонстрированы базовые приемы работы с библиотекой для передачи уведомлений клиентам web-приложений. После прочтения данной статьи читатель может приступить к более глубокому изучению библиотеки, применяя информацию на практике. Список статей для дальнейшего изучения приведен ниже.

### **Список использованных источников**

1. Блог Александра Кондуфорова <http://merle-amber.blogspot.ru/>
2. Интернет ресурс <http://signalr.net/>
3. Интернет ресурс <https://github.com/SignalR/SignalR/wiki/>